CS413 Lab 1

ARM Program using Auto-Indexing - ADD

Objective: The objective of this lab is to have the students use the ARM Auto-Indexing to access array elements and to do nested subroutine calls. Reference text book sections 3.7, 3.8 and 3.10.

Write an ARM assembly program that performs the following:

1. Declare three arrays that will contain integers. Arrays one and two will be initialized in the code. The values of arrays will contain integers such that when array elements are added the results are positive, zero and negative.
2. Array three will be calculated to be equal to the Array one[i] + Array two[i] and must be performed using a for loop construct and auto-indexing.
3. The program is to print a welcome message with instructions to the user.
4. The program is to print the elements of all three arrays with text on which array is being printed.
5. The array print must be performed by a subroutine and accessed by using the ARM Assembly instruction BL. The starting address of the array to be printed will be passed into the subroutine. The address may be passed via a register or the stack via the PUSH and POP commands. Since this print subroutine is not a leaf routine code must be in place to properly save and restore the link register (r14).
6. The array sizes have to be 10 elements.
7. The user will be prompted to enter a positive, negative or zero. The software will then print only the numbers of that type (positive, negative, zero) from array three. The software must use a for loop and auto-indexing to access the array elements.
8. Assume all inputs are valid and no overflow conditions will occur.

The GTA will review your assembly code to ensure you are using the required Auto-Indexing modes and printing of arrays is performed using a subroutine

The general outline for your code should look like:

```
main:
     Print welcome message
     For i=1 to 10 sumarray[i] = array1[i] + array2[i];
     Printarray(array1)
     Printarray(array2)
     Printarray(sumarray)
     Use scanf to get user to enter positive, negative or zero.
     For i=1 to 10 if sumarray[i] matches user input sign type
     print sumarray[i];
End main

Function Printarray(inarray)
```

```
     For i = 1 to 10 print inarray[i]
End function Printarray
```

Prework:

Students need to have a file which, at the very least contain the following:

1. Proper header information that include commands that assemble, link, run the file and run the file using the debugger that is specific for their file.
2. Data section that defines all the output strings.
3. Data section that defines the three arrays. Two arrays are initialized with zero, positive and negative integer values.
4. Code comment sections that describe the major sections of the code with the functions for the print array clearly identified.

This lab was assigned

Spring 2020

Fall 2021

CS413 Lab 1

ARM Program using Auto-Indexing - ADD

Student ________________________ Lab Instructor: ________________________

Total Score: ________________ Late Penalty? No _____ Yes: ______

Feature	**Points off**
Code Comments/Documentation ● Class, Term, Author, Date. ● Purpose of software ● Documentation for the start of each loop and function (subroutine) ● Documentation for error checking ● In-line comments that explain why the following code exists.	
Review code to ensure: ● Array 1 and 2 are declared in the code ● Auto-indexing and a loop is used to calculate array 3 ● Subroutine is used to print arrays and the starting address of the array is passed into the subroutine. ● For loop and auto-indexing is to print user selected numbers from array 3	
Welcome and instruction message are displayed and clear.	
All three arrays are printed and match what is defined and calculated in the code.	
Enter positive and ensure only the positive numbers from the calculated array is printed.	
Enter negative and ensure only the negative numbers from the calculated array is printed.	
Enter zero and ensure only the zero numbers from the calculated array is printed.	
Note: It is not a requirement for the program to loop back to prompt for user inputs. The code can ask for input, print results then exit.	